

TABLE NO

--	--	--

STUDENT ID NO



--	--	--	--	--	--	--	--	--	--

SUBJECT CODE _____

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 2, 2018/2019

TSE3151/TSD2711 – SOFTWARE DESIGN
(All sections / Groups)

13 MARCH 2019
2:30 pm – 4:30 pm
(2 Hours)

Examiner 1 Signature: _____

Examiner 2 Signature: _____

Examiner 3 Signature: _____

Question	Mark
A	
B	
C	
D	
Total	

INSTRUCTIONS TO STUDENTS

1. This question paper consists of 11 printed pages (including cover page) with 4 Sections only.
2. Attempt **ALL** questions in **SECTION A**, **SECTION B**, **SECTION C** and **SECTION D**. The distribution of the marks for each question is given.
3. Please write all your answers **CLEARLY** in the specific answer box provided for each question. Submit this question paper at the end of the examination.

Attempt ALL questions in SECTION A, B, C and D.

Section A (12.5 marks)

Consider the following **SingletonPatternEx.java** program.

```
// SingletonPatternEx.java
package singleton.pattern.demo;

class MakeACaptain
{
    private static MakeACaptain _captain;
    //We make the constructor private to prevent the use of "new"
    private MakeACaptain() { }

    private static class SingletonHelper{
        //Nested class is referenced after getCaptain() is called
        private static final MakeACaptain _captain = new MakeACaptain();
    } //end class SingletonHelper

    public static MakeACaptain getCaptain()
    {
        return SingletonHelper._captain;
    }
} //end class MakeACaptain

class SingletonPatternEx
{
    public static void main(String[] args)
    {
        System.out.println("***Singleton Pattern Demo***\n");
        System.out.println("Trying to make a captain for our team");
        //Put your code here
    }
}
```

Based on the above context, answer the following questions Q-A1 to Q-A3:

A1. Rewrite the main method of SingletonPatternEx.java to produce the following output by creating two instances of singleton, and comparing whether they are the same instance.

F:\YourName>javac singleton\pattern\demo\SingletonPatternEx.java

F:\YourName>java singleton.pattern.demo.SingletonPatternEx

Singleton Pattern Demo

Trying to make a captain for our team

Trying to make another captain for our team

c1 and c2 are same instance

(5 marks)

Continued...

Answer

A2. Draw a Class Diagram to show TWO (2) relationships within these two classes, plus the attributes and methods.

(4.5 marks)**Continued...**

A3. There are TWO (2) situations on what do we need to do to incorporate thread safety in the above Singleton implementation.

A3a. For the first situation, explain how we use of the “synchronized” keyword, and its disadvantage.

(1 mark)

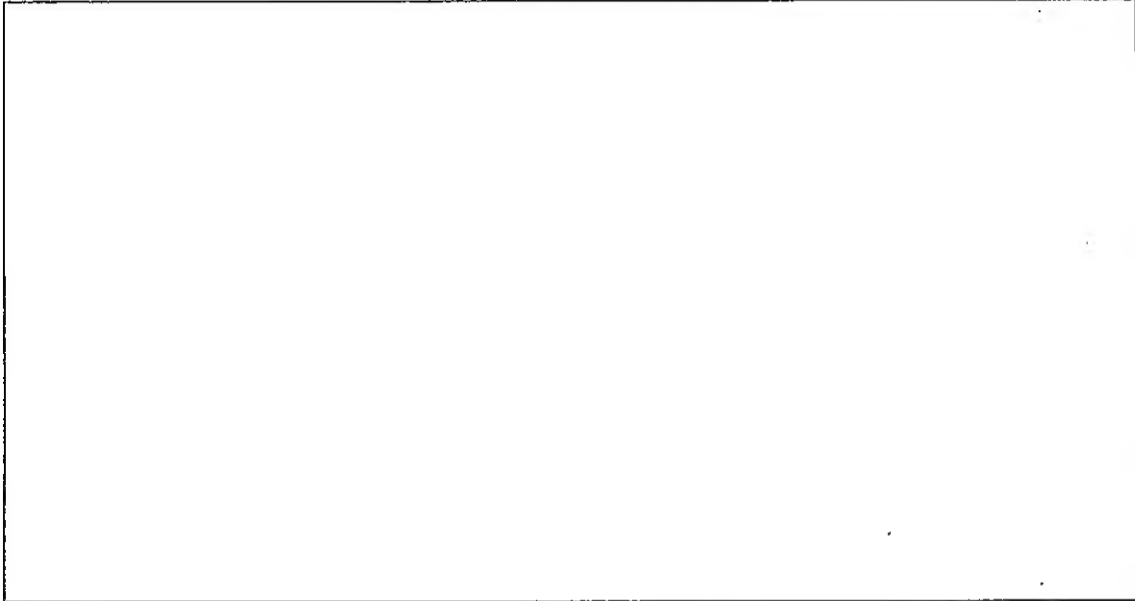
A3b. For the second situation, explain the THREE (3) steps of eager initialization of class MakeACaptain to achieve thread safety. What is its advantage?

(1.5 + 0.5 mark)

Continued...

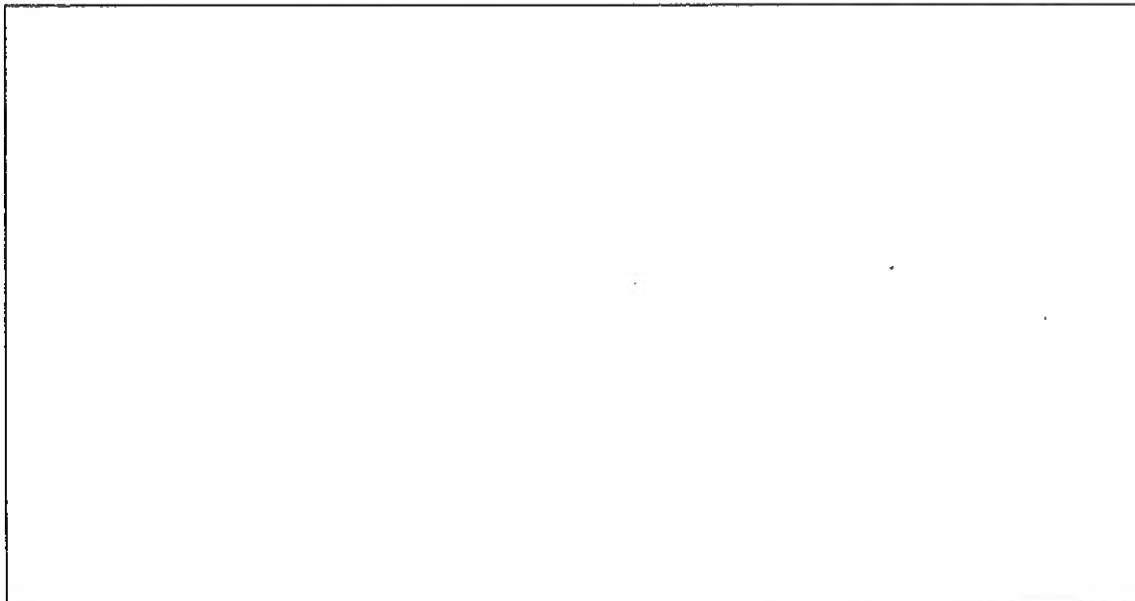
B2a. Designing software is made more complex because we may be designing a sequence of actions. Design for a set of instructions for making tea with a teapot and teabags.

(4 marks)



B2b. Consider the major problems that might arise (examples: no water in the kettle, burst teabag, no kettle and so on). How would you reorganize the instructions in your answers to question B2a above, for these exceptional situations so that they do not obscure the original design?

(2.5 marks)



Continued...

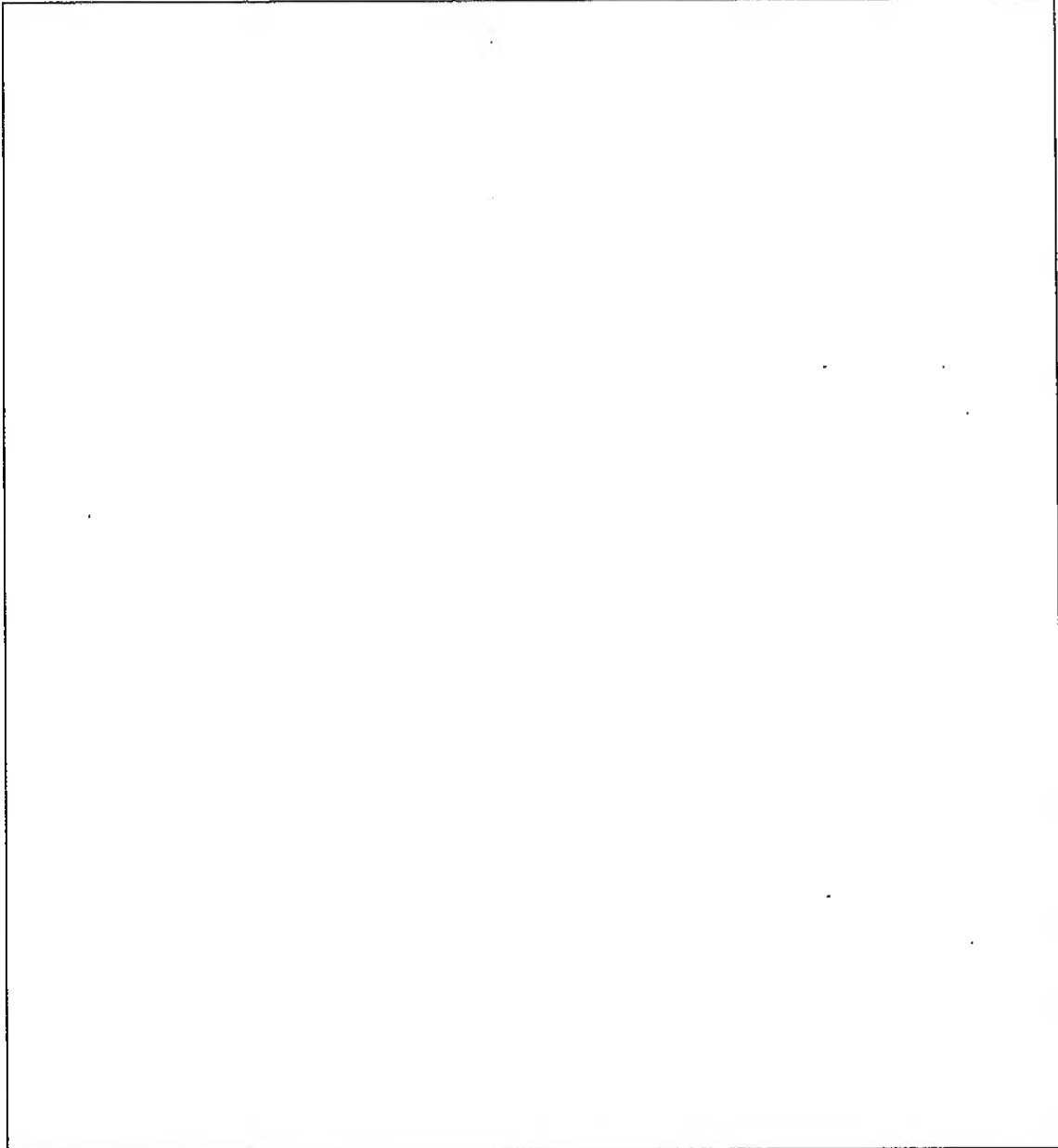
Section C (12.5 marks)

C1. Consider a physical tower electrical grid design used with a software aided design system. Describe the following:

(C1a) THREE (3) viewpoints that might be needed in order to provide a full design description,

(C1b) FOUR (4) representations with examples that could be used for these design descriptions.

(3 + 2 marks)



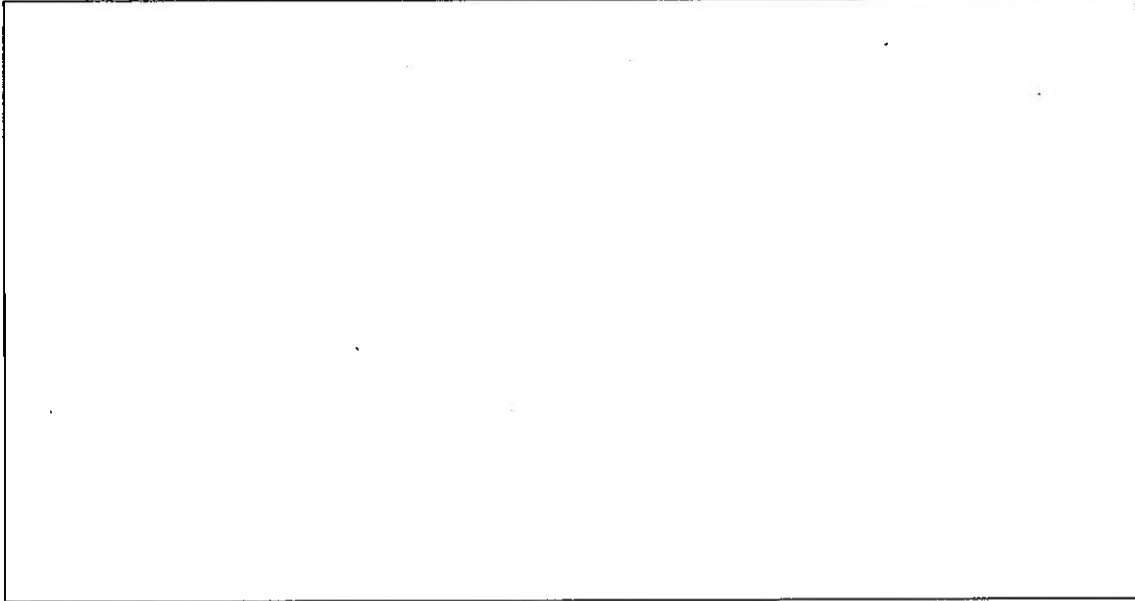
Continued...

C2. Write down and explain:

(C2a) TWO reasons in favour of standardizing any particular form of design description;

(C2b) THREE reasons against standardizing the same form of description.

(2 + 1.5 marks)

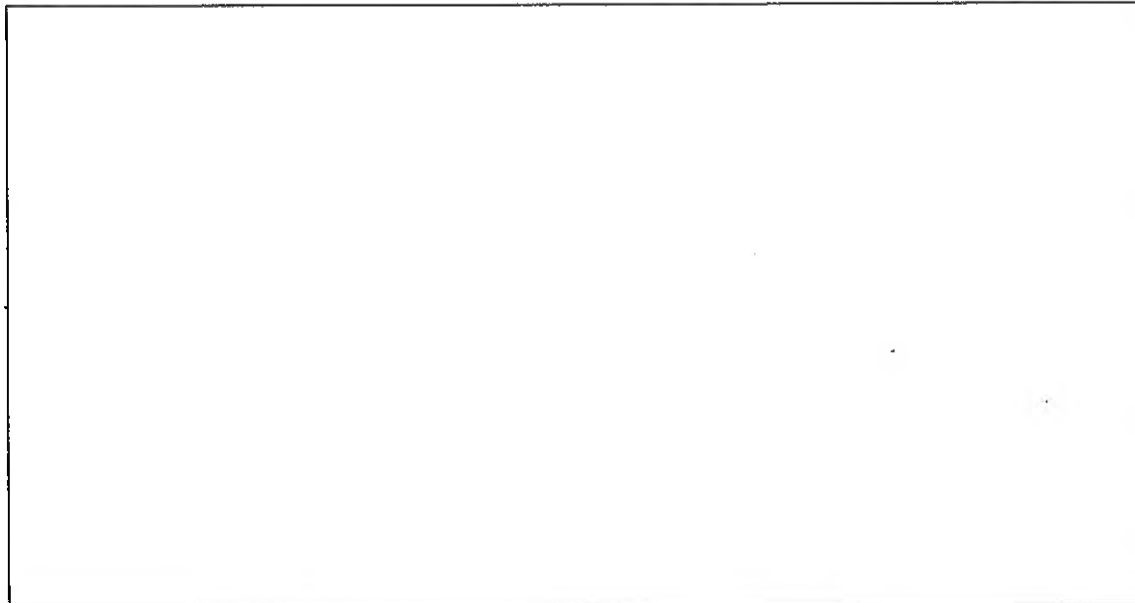


C3. Suggest how you might represent the following viewpoints using in turn: text on its own; and a diagram:

(C3a) the hierarchy of pages in a website;

(C3b) the program units (procedures) that make use of a particular data type in a program.

(2 + 2 marks)



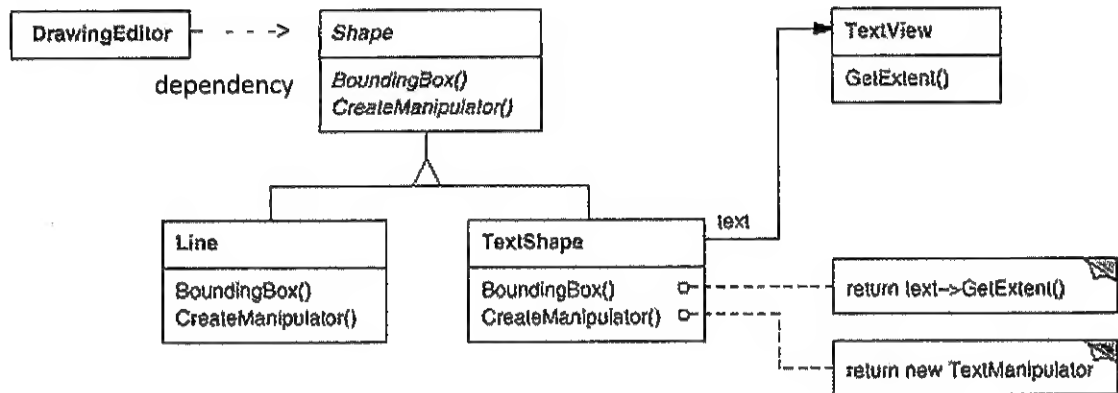
Continued...

Section D (12.5 marks)

Among the name in the **Adapter** Design Pattern (DP) include *Adaptee*, *adapteeMethod()*, *Target*, *targetMethod()*, *Adapter*, *adapteeAggregationVariable*, and *Client*.

Based on the above context, answer the following questions Q-D1 to Q-D2:

D1. Write a table to show the mapping of **Adapter** DP to TextShape and TextView Motivation Example given in the following class diagram.

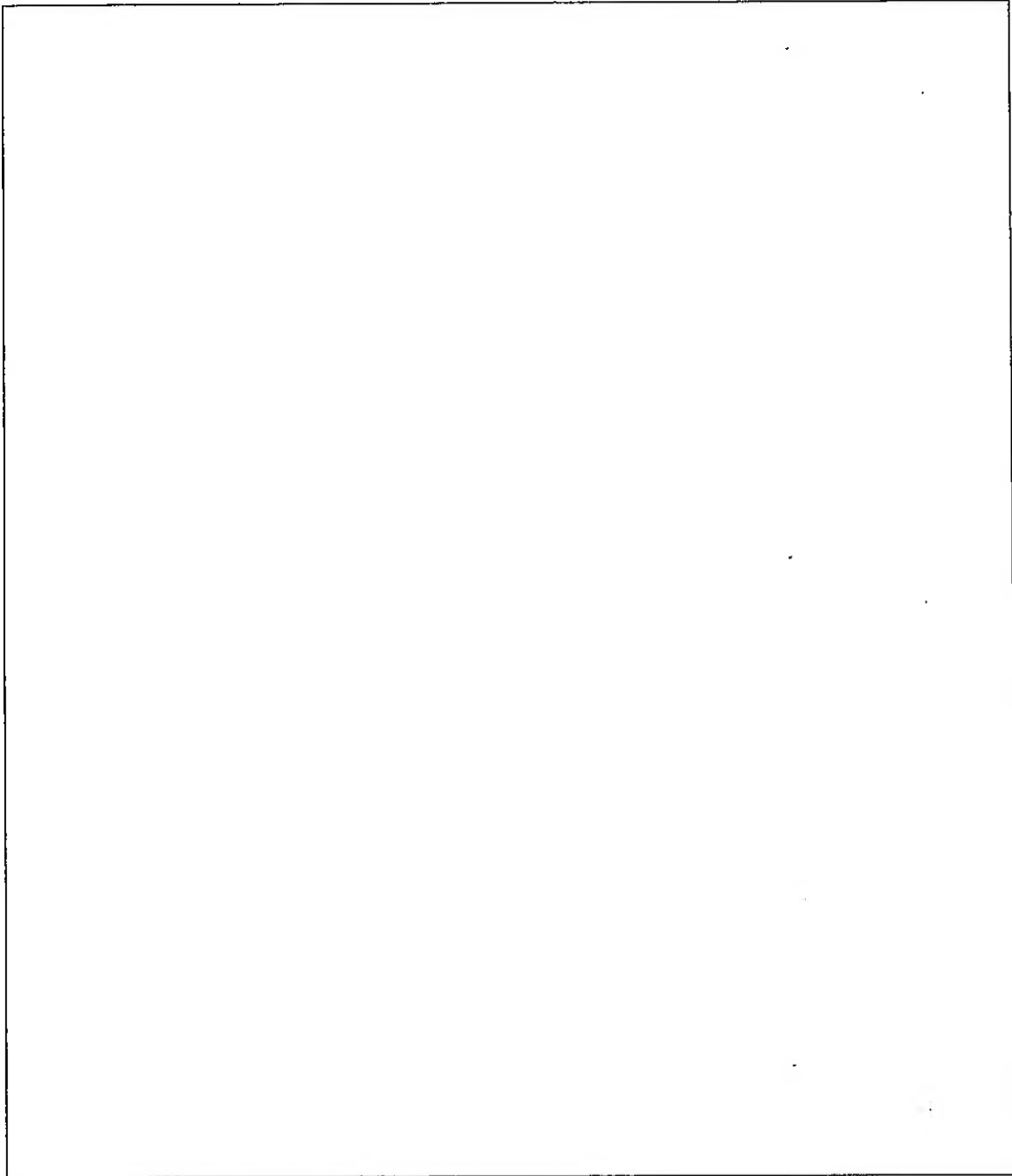


(4 marks)

Continued...

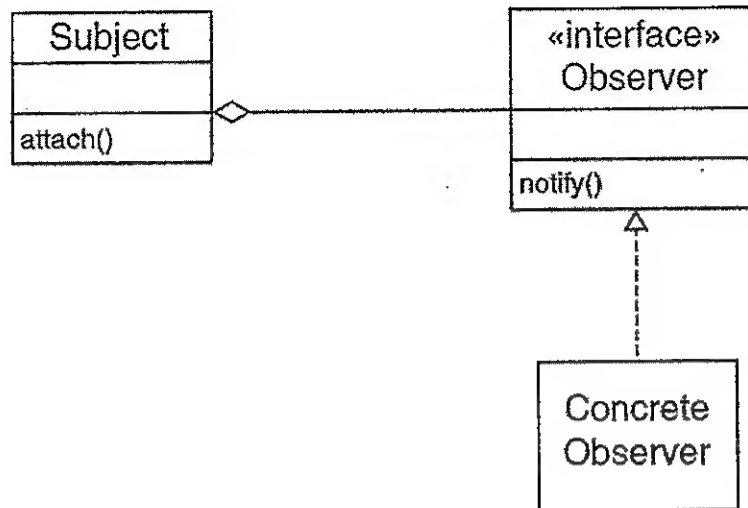
D2. Draw a class diagram for the typical Adapter DP with the name in DP of *Adaptee*, *adapteeMethod()*, *Target*, *targetMethod()*, *Adapter*, *adapteeAggregationVariable*, and *Client*.

(5 marks)



Continued...

D3. Write a table to show the mapping of **Observer** DP given in the following class diagram to the case of JButton and its event listeners. Among the actual name include JButton, addActionListener(), actionPerformed(), ActionListener, and the class that implements the ActionListener interface type.



(2.5 marks)

--

D4. Design patterns are a form of reuse. What are you reusing in design patterns?

(1 mark)

--

End of Papers